Faculdade de Engenharia da Universidade do Porto

Food Supplement Personal Assistant

Nuno Miguel dos Santos Castro

Mestrado Integrado em Engenharia Informática e Computação

July 23, 2019

Food Supplement Personal Assistant

Nuno Miguel dos Santos Castro

Mestrado Integrado em Engenharia Informática e Computação

Approved in oral examination by the committee:

Chair: Carla Alexandra Teixeira Lopes
External Examiner: Mário Jorge Ferreira Rodrigues
Supervisor: Liliana da Silva Ferreira
July 23, 2019

Abstract

Nutritional supplements market is increasing worldwide. In 2015, the global supplements market was €109 billion and is expected to grow to €160 billion by 2020. More and more persons need help managing their diets and nutritional intake. The use of technology can increase nutritional education and improve the security of the treatment. Diet-related health anomalies are becoming more frequent not only in developing countries but also in the westernized society.

In order to address this problem, a system that interacts with the user, via a mobile application, and recommends nutritional supplements was created. This system analyzes the supplement's label statements, and automatically extracts information regarding the effects of the supplement on the user's health.

This system makes use of the Dietary Supplement Label Database. This dataset is composed of 65,684 different supplements, that are used for recommendation generation. For a set of 50 supplements and their respective recommendations, with the very low threshold of 10%, the system achieved a precision value of 0,39; recall value of 1 and f1-score of 0,57.

From the results it is possible to conclude that this system has some limitations regarding the extraction of relevant information from the supplement statements. It is also possible to infer that the usage of more structured information can increase the performance of such extraction.

Resumo

O mercado da suplementação nutricional está a crescer mundialmente. Em 2015, o mercado global de suplementação nutricional era de €109 mil milhões e espera-se que cresça para €160 mil milhões. Cada vez mais pessoas necessitam de ajuda a gerir as suas dietas e os nutrientes que consomem. O uso da tecnologia pode melhorar a educação nutricional e melhorar a segurança de tratamento. Anomalias relacionadas com a alimentação estão a tornar-se cada vez mais comuns, não só nos países em desenvolvimento mas também na sociedade ocidental.

De maneira a resolver este problema, foi criado um sistema que interage com o utilizador, através de uma aplicação móvel, e recomenda suplementos nutricionais. O sistema analisa as instruções da etiqueta do suplemento, e automaticamente extrai informação relativa aos efeitos do suplemento na saúde do utilizador.

O sistema utiliza a base de dados Dietary Supplement Label Database. Este conjunto de dados é composto por 65.684 suplementos diferentes, que são usados para a geração de recomendações. Para um conjunto de 50 suplementos e as suas respetivas recomendações, com um valor de *threshold* muito baixo de 10%, o sistema conseguiu alcançar um valor de *precision* de 0,39; valor de *recall* de 1 e valor de *f1-score* de 0,57.

A partir dos resultados é possível concluir que este sistema possui algumas limitações quanto à extração de informações relevantes das instruções do suplemento. Também é possível inferir que o uso de informações mais estruturadas pode melhorar o desempenho da extração.

Acknowledgements

First of all, I would like to thank my supervisor Liliana Ferreira and researcher Diego Silva, for their help and guidance throughout the development of this dissertation.

I would also like to thank all my colleagues at Fraunhofer AICOS Portugal, for making me feel at home and providing help and counseling when it was needed.

I also want to thank all my friends for putting up with my bursts of craziness, for keeping me motivated, giving me the support I needed to finish this important step of my life, and for not allowing me to go crazy over these last six intense months.

Last but not least, I want to express my sincerest gratitude for my parents and grandparents, for providing me with the emotional and financial support to finish my Master's Degree.

Nuno Miguel dos Santos Castro

"Science may set limits to knowledge,
but should not set limits to imagination."

Bertrand Russell

Contents

List of Figures

List of Tables

Abbreviations

CBRS	Content-based recommendation system
CRS	Collaborative recommendation system
CVS	Cosine Vector Similarity
DSLD	Dietary Supplement Labels Database
HRS	Health recommendation system
HyRS	Hybrid recommendation system
IA	Intelligent agent
MAE	Mean absolute error
MBRS	Memory-based recommendation system
MoBRS	Model-based recommendation system
N3	Notation 3
NBRS	Neighborhood-based recommendation system
NLP	Natural language processing
OWL	Web Ontology Language
PCC	Pearson's Correlation Coefficient
PDA	Personal digital assistant
PSM	Problem Solving Methods
RDF	Resource Description Framework
RDFS	Resource Description Framework Schema
W3C	World Wild Web Consortium

Chapter 1

Introduction

1.1 Context and Motivation

Nutritional supplements market is increasing worldwide. By 2020, the global market is expected to grow to €160 billion [2]. It is becoming increasingly simple to access dietary supplements that are expected to improve or prevent various conditions. Many nutritional supplement users obtain them without the consent or recommendation of a medical practitioner, which may be problematic when misused [3]. Medical assistance regarding nutritional supplementation is still being rejected, however, informing the general public about their supplement intake is still of utmost importance. Additionally, retailers and great supermarkets experience a growing interest in this type of supplements, being interested in tools that can support buyers in their choices.

With the support of Fraunhofer Portugal, this dissertation will research the best approach to dietary supplement recommendations, as well as the most appropriate choice for natural language processing algorithms to develop a system capable of informing and recommend nutritional supplements to its users.

1.2 Project Objectives

This dissertation has three different objectives.

Firstly, different health and nutritional recommendation systems are explored. By knowing the available technology regarding these systems, it becomes easier to select and develop the correct type. Further in Chapter 2, the different types of recommendation systems will be addressed.

The second objective consists on exploring and creating nutritional knowledge bases to use together with the previously defined recommender system. In Chapter 3 the ontology will be created and the dataset defined.

The last goal consists on creating an application that joins the knowledge base with the recommendation system. It is expected that this application will be able to provide accurate recommendations of what nutritional supplements the user should consume.

1.3 Contributions

The findings of this thesis will help the further development of intelligent systems regarding nutritional supplements. The growing interest by the retailers and large supermarkets on these types of tools, as well as the consumers necessity for information, justify the need for the creation of a system that can inform and suggest these products. The creation of an ontology structure for nutritional supplements can further aid the development of more tools like the one this thesis proposes, as none was previously available.

Finally, the created recommendation system can be utilized as an extra module for a variety of applications that might be related to nutritional supplementation, or even other recommender engines related to the consumption of healthier food groups.

1.4 Document Structure

This document is divided in six main chapters.

First and foremost, Chapter 2 presents the state of the art of the domain, including a brief review of health intelligent virtual assistants. In this survey, intelligent assistants are described, and some existing e-health applications related to nutrition are presented. Later in this chapter, different techniques regarding recommendation systems and natural language processing are explored.

Secondly, in Chapter 3, a brief theoretical introduction on ontologies is provided. Furthermore, the nutritional supplement ontology structure is presented, as well as the methods used to collect all the information on nutritional supplements.

Subsequently, the methodology used to develop this project is presented in Chapter 4. The decisions regarding the employed database in the system in addition to the type of recommendation engine used are explained. Finally, the mobile application structure is provided as well as a quick overview of the system's architecture.

Afterwards, the validation of the recommendations and the application itself is given. Later in Chapter 5, the project results are presented and discussed.

Last but not least, Chapter 6 will give the conclusions of this project in addition to present the possible future work that could be done.

Chapter 2

Background

This chapter will focus on the presentation of the different available methodologies regarding recommendation systems and natural language processing. It will begin with a concise contextualization about e-health applications, followed by a brief overview of the different types of recommendations systems. Finally, the different techniques used in natural language processing will be presented.

2.1 Health intelligent virtual assistant

2.1.1 Intelligent Agents

As Don Gilbert states, "an intelligent agent is software that assists people and acts on their behalf." [4]. These intelligent agents (IA) work as assistants to the user, automating monotonous tasks such as data summarization and recommendations. Consequently, IA can also be called personal digital assistants (PDA). IA have three primary tasks that include understanding changing conditions, react upon them and then take actions in order to conclude what should be done [5]. An example of the IA's tasks is shown on the Figure 2.1.

IA have been used to manage a wide range of health related problems, such as decision support or knowledge base systems [7, 8].

The employment of e-health decision support tools, such as the aforementioned PDA's, can successfully change a patient's behavior regarding its health care and welfare, for instance, by supporting its decision regarding nutritional supplements.

2.1.2 E-health

E-health is a concept that is used to represent anything related to computers and medicine. This term appeared when industry and marketing leaders attempted to bring e-commerce

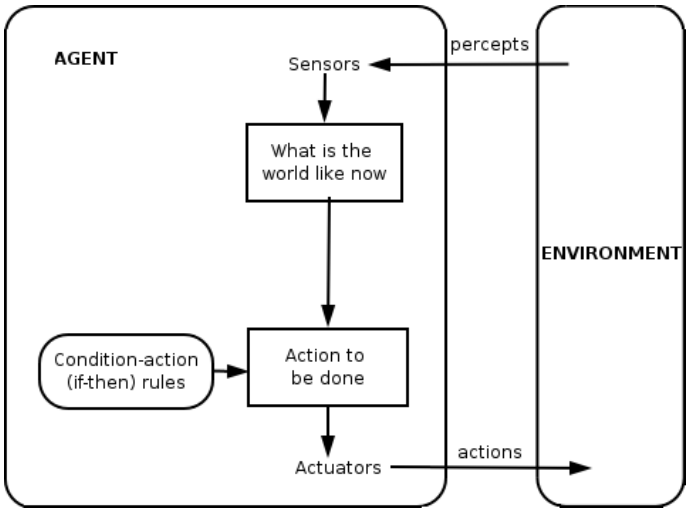

Figure 2.1: Example of an intelligent agent [6].

principles into the health sector, as well as depict the new opportunities the Internet could introduce to the same area. E-health can then be defined as a field that utilizes medical informatics, public health and business to improve health care and welfare, by utilizing information and communication technology [9].

The term information overload was introduced by Bertram Gross in 1964, in his work - The Managing of Organizations. It is used to describe the difficulty to process a problem and make decisions when there is a huge amount of input, as shown in Figure 2.2 [10].

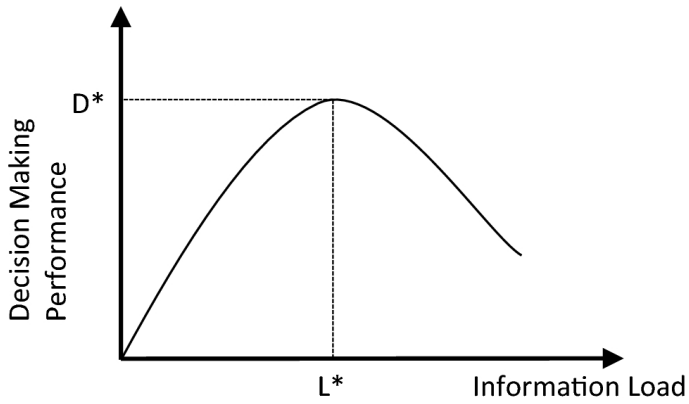

Figure 2.2: Information and decision-making performance.

The problem of information overload can be overcome by employing a recommendation filtering technique. This methodology allows the filtering of unnecessary data and, in a proactive manner, present relevant information to the user [11, 12, 13].

2.2 Recommendation systems

A recommendation system is an engine that filters information in order to prevent data overload. By filtering fragments of the information from the input, the system can improve the decision making process and quality. This procedure can be further optimized by taking into account the user demographics while filtering the output [14, 15].

The recommendation system process is shown in Figure 2.3, with its key phases listed as follows:

- **Information collection** - This phase is responsible for collecting user data in order to generate a model for the prediction tasks. This gathered data includes user demographics or external resources accessed by the same user. There are three types of information gathering approaches. These are explicit, implicit and hybrid feedback.

    1. **Explicit feedback** - The user explicitly rates items so that his model is constructed and improved. The downside of explicit feedback is that it depends on the information provided by the user, which may not be in the required quantity. However, this method is still one of the most reliable data providers as it deals directly with the user actions and does not interact with user demographics.

    2. **Implicit feedback** - The system deduces user's preferences through its actions, i.e. purchase and navigation history, button clicks, etc. This method requires no effort from the user whatsoever, but it's accuracy is lower.

    3. **Hybrid feedback** - This method applies the advantages of both methods in order to reduce their weaknesses and increase the prediction accuracy. Implicit information is used as a verification on explicit data given by the user.

- **Learning** - An algorithm is utilized to extract features from the feedback gathered in the previous phase.

- **Prediction** - This phase predicts the user preferences regarding the items. Can be made directly on the collected data or through the user's observable actions. The prediction can be made in one of two ways: either by taking into account the information collected previously or through the observed user actions.

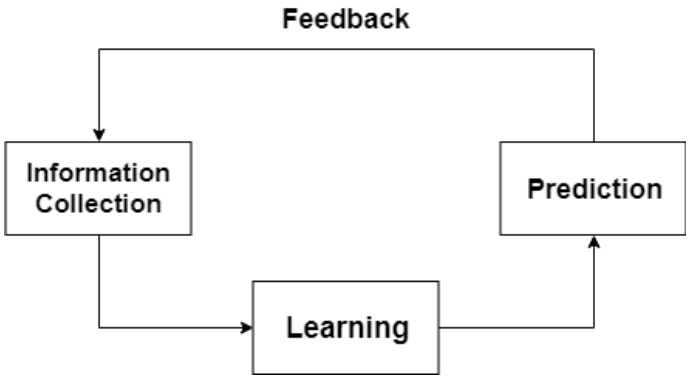

Figure 2.3: Recommendation process [14].

2.2.1 Content-based recommendation systems

Content-based recommendation systems (CBRS) utilize an algorithm that takes into account item descriptions in order to determine relevant items to the user [16]. The high-level architecture of the CBRS is represented on Figure 2.4. There are different types of data representations:

- **Structured data** - This representation contains a small number of item characteristics. Each is described by the same set of fields, with some of those having a known set of possible values. For instance, a relational database model is considered structured data.
- **Unstructured data** - This representation does not have a defined data model, unlike the structured data representation. An example of unstructured data is an email: it has a subject, recipient and a sender but its body is unstructured text.
- **Semi-structured data** - This representation contains unstructured data, however it has information associated with it, such as keywords. An example of semi-structured data is markup languages.

There are two different recommendation approaches currently being adopted in CBRS. One is treating the problem as an information retrieval task, by handling the user's preferences as a query and then scoring the documents with relevance to this query. The other approach is treating the problem as a classification task, where the user's past ratings are used as labels to classify the content of an item. There are several different algorithms employed in the classification approach, such as Naïve Bayes Classifier, K-Nearest Neighbor and Artificial Neural Networks [17].

2.2.2 Collaborative recommendation systems

Collaborative recommendation systems (CRS) build a database of preferences of items by users. This database is called an user-item matrix. It then calculates similarities between

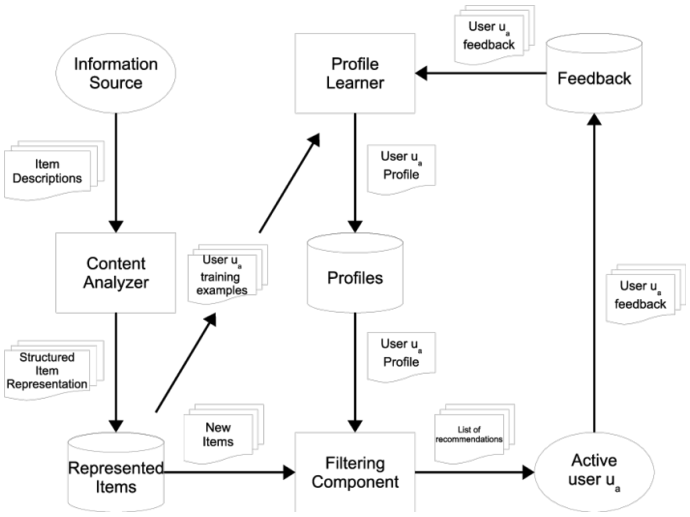

Figure 2.4: Content-based filtering process [18].

user profiles and matches them based on those similarities, creating groups of users called neighborhoods. Recommendations are made to the user based on the positive ratings other users in the neighborhood made [14]. The recommendation process can be seen in Figure 2.5

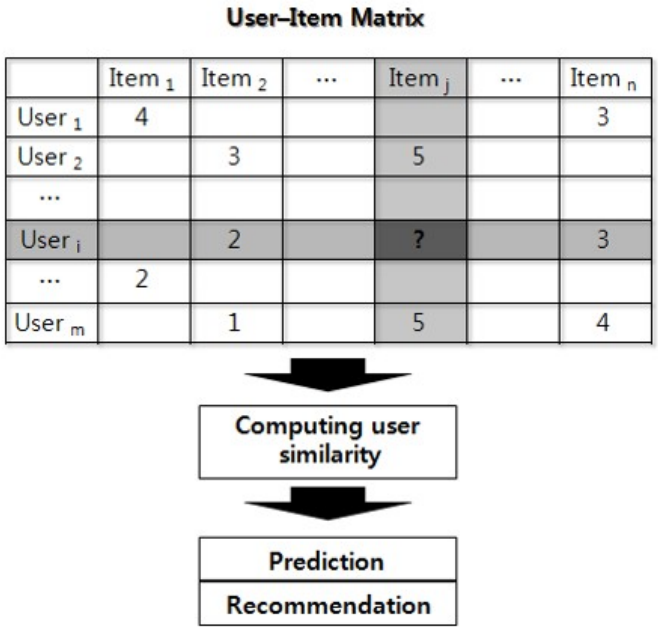

Figure 2.5: Collaborative filtering process [19].

CRS can be divided into two different categories: memory-based and model-based.

Memory-based

Memory-based recommendation systems (MBRS), also known as Neighborhood-based recommendation systems (NBRS), involve three steps:

1. **User similarity measurement** - This step quantifies the similarity weight $w_{x,y}$ between user x and user y, taking into account their ratings on items shared by both individuals. There are several similarity measures available, and among them, the Pearson's Correlation Coefficient (PCC) and Cosine Vector Similarity (CVS) are the most common [20].

   PCC is calculated by the following formula, where $r_{x,i}$ represents an item rating given by a user. Additionally, $I_{x,y}$ represents an item group with ratings from both users x and y.

   $$r = \frac{\sum_{i=1}^{n}(x_i - \overline{x})(y_i - \overline{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \overline{x})^2(y_i - \overline{y})^2}}$$

   CVS is calculated by the following formula, where $x \cdot y$ corresponds to the dot product between the rating vectors of user x and user y respectively. Furthermore, $||x||$ and $||y||$ represent the vector's x and y length, respectively.

   $$cos(x, y) = \frac{x \cdot y}{||x|| \cdot ||y||}$$

2. **Neighborhood selection** - There are two different methods while selecting a neighborhood for the user: similarity thresholding and best-N-neighbors. The first method employs a threshold L to filter out users with similarities that are below L. The last method chooses the top N most similar users to the active one.

3. **Prediction generation** - The last step consists on aggregating ratings on a certain item given by all the users in the neighborhood and the similarities calculated in the user similarity measurement step. The most common form of aggregation is depicted in the following formula:

   $$p_{a,i} = \overline{r_a} + \frac{\sum_{x \in N} w_{a,x} x (r_{x,i} - \overline{r_x})}{\sum_{x \in N} |w_{a,x}|},$$

   where $p_{a,i}$ represents the preference for the active user a on an item i, and $\overline{r_a}$ and $\overline{r_x}$ represent the mean ratings for all rated items done by the active user a and user x respectively.

There are two types of MBRS: user-based and item-based. Both types operate on an user-item matrix. The difference between them is that, in the user-based approach, the algorithm rates an item i by an user u by joining the ratings of other users similar to u,

while in the item-based approach, the algorithm produces a rating for an item i by an user u by analyzing the set of items i' that are similar to i that u has rated and then combines the ratings by u of i' into a predicted rating by u on i [21].

Model-based

Model-based recommendation systems (MoBRS) are a type of CBRS that generate recommendations by roughly calculating parameters of statistical models for user ratings [14].

Recently, the most common methodologies used are latent factor and matrix factorization models. According to Melville, "latent factor models assume that the similarity between users and items is simultaneously induced by some hidden lower-dimension structure in the data" [17]. Matrix factorization techniques characterize both users and items by vectors of factors deduced from patterns in item ratings, with respect to some loss measure [22]. The standard choice for the loss function is the squared loss.

2.2.3 Hybrid recommendation systems

It is known that CBRS and CRS have limitations. However, the hybrid recommendation systems (HyRS) combine both approaches in order to increase the accuracy and effectiveness of the recommendations. By using various techniques, it becomes possible to decrease the number of drawbacks of an individual system. Table 2.1 shows the different methods utilized with a brief description.

Method	Description
Weighted	Combines results from different recommenders to create a recommendation list.
Switching	System switches one of the recommendation techniques.
Cascade	Recommender applies refinement process over the recommendations given by another.
Mixed	Instead of having only one recommendation per item, several are presented at the same time.
Feature-Combination	Characteristics from different recommendation techniques are combined in a single algorithm.
Feature-Augmentation	Characteristics produced by a recommendation technique are used as input for another.
Meta-level	The model created in a recommendation technique is utilized as an input for another.

Table 2.1: Different hybridization methods [12]

2.2.4 Evaluation metrics

A recommender system's quality can be evaluated using different metric types. Depending on the filtering technique, a different kind of metric can be used. Metrics can be categorized into two main classes:

- **Statistical accuracy** - Evaluates the system's accuracy by collating the recommendation scores with the user ratings for the user-item pairs in the test

database. The most common metric used is Mean Absolute Error (MAE), which is a measure of difference between recommendations from their true user-specified values. MAE is given by the following formula:

$$MAE = \frac{\sum_{i=1}^{N} |p_i - q_i}{N},$$

where $<p_i, q_i>$ is the ratings-prediction pair and N is the number of corresponding pairs.

A recommendation system is most accurate when the MAE is lowest. Besides MAE, Root Mean Squared Error (RMSE) and Correlation can also be used as a metric for statistical accuracy [23].

- **Decision support accuracy** - Evaluates the effectiveness of predictions at assisting a user select relevant items. The prediction process is assumed to be a binary operation. The most common metrics used are reversal rate, weighted errors and ROC sensitivity [23].

2.2.5 Health recommendation systems

Health recommendation systems (HRS), which are a part of recommendation systems applied in the health sector, are an example of e-health applications. HRS's target audience can be divided into two groups: physicians which use HRS as an assistant during diagnostics; and patients which use HRS as a personal health advising tool in their welfare [24]. The latter group focuses on delivering trustworthy information to patients on different domains, for instance food and nutritional information; behavior change recommendations such as suggesting on how to improve eating, exercising or sleeping habits; and increase patient safety [25].

FitGenie [26] and *Fooducate* [27] are good illustrations of HRS. The first is a calorie counter that makes adaptive macro nutrient adjustments to help the user achieve its dietary goals while the second scans and grades food, and presents recommendations for healthy alternatives. However, none of these applications provide recommendations on nutritional supplementation.

2.3 Natural language processing

Any language that is used for communication between humans, can be considered a natural language. In order to interact with computers using natural language, some processing must be done.

Natural language processing (NLP) is a branch of artificial intelligence that manipulates, understands and interprets natural language in order to interact with humans. NLP has a variety of applications, some of which are information retrieval

and extraction, document summarization, question answering and machine translation [28, 29, 30, 31]. In Figure 2.6, the different steps of NLP are shown.

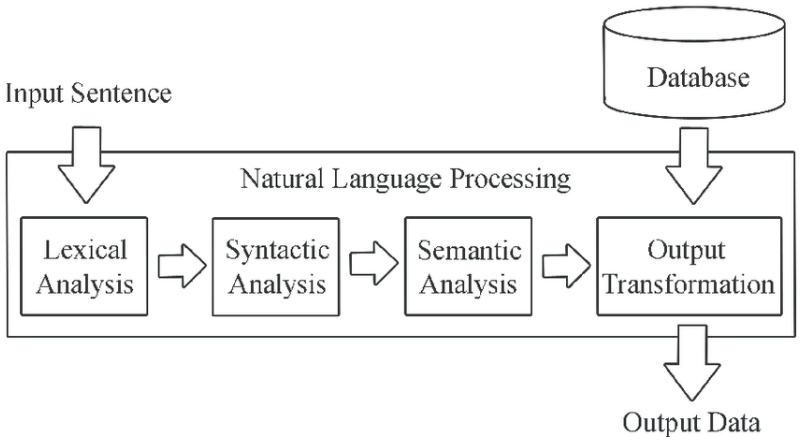

Figure 2.6: Natural Language Processing step sequence [32].

Lexical Analysis

This step analyzes the textual input and connects each word with a corresponding label in a dictionary. One of the main difficulties of this step is when a word has more than one meaning, which can the process of choosing the correct meaning of the word impossible. One of the many applications of lexical analysis is to help on the prediction of the grammatical function of words that were previously unknown [33].

Syntactic Analysis

Syntactic analysis processes a sentence syntactically, determining the subject, predicate and the location of, for example, nouns in the sentence. However, several problems may occur: (1) a word can function as different roles of speech in distinct contexts, (2) a sentence may have opposing meanings depending on the interpretation, (3) there are several ways of interpreting a sentence [34].

Semantic Analysis

Semantic analysis refers to the analysis of meaning of words and sentences [35]. It processes the surrounding text and analyzes the overall text structure in order to accurately find the proper meaning of words that have more than one connotation. For example, a text about "politics" and "economics", if semantically analyzed, will relate to concepts such as "elections" or "budget" and "tax".

Output Transformation

In this step, the main objective is to process the obtained information from the last steps into an interpretable source in order to assist the decision making procedure.

2.3.1 Information extraction

There are some feature extraction algorithms utilized in NLP. One of such algorithms is the Term Frequency-Inverse Document Frequency (TF-IDF) [36]. TF-IDF identifies the occurrence ratio of terms in a document compared to the inverse ratio of that term in the entire document corpus. This procedure calculates the relevance of a word in a document. A very common term, such as articles or prepositions, tend to have lower scores than terms that appear only in a single or a small group of documents. The score is calculated by the following formula:

$$w_d = f_{w,d} * d * log(\frac{|D|}{f_{w,D}}),$$

where $f_{w,d}$ is the number of times the word w appears in the document d, $|D|$ is the size of the document corpus and $f_{w,D}$ equals to the number of documents in which the word w appears.

2.3.2 Document summarization

Document summarization aims to create a brief and fluid summary that expresses the main ideas of a text [37]. There are three main tasks that a summarization system has to perform:

1. **Intermediate representation** - This representation allows the systems to identify relevant information.

2. **Score sentences** - After the definition of the intermediate representation, each sentence is assigned a score that represents its relevance in the text.

3. **Select summary sentences** - Once every sentence has a score assigned, the summarizer must select the best combination of sentences to form a summary.

TextRank model for document summarization

TextRank [38] is a graph-based ranking model derived from Google's PageRank [39], that can be employed in various natural language processing applications. Graph-based ranking algorithms allow the decision of the importance of a vertex within a graph, based on global information recursively extracted from the whole graph.

TextRank can be utilized for sentence extraction for automatic summarization of text documents. The main goal is to rank entire sentences, so, for each sentence in the text,

a vertex is added. Then, by estimating the similarity between vertexes, a connection can be formed between two vertexes that share similarities. This similarity can be calculated using the cosine similarity measure. Finally, by applying a weighted graph-based ranking formula in the graph, it becomes possible to select the top-ranked sentences for insertion in the summary.

2.3.3 Sentiment analysis

Nearly every human activity is influenced by beliefs. Our decisions and the way we perceive the world are somewhat affected by how others evaluate the world. Thus, our decision-making process is susceptible to the opinions of others.

Sentiment analysis assembles distinct fields of research, like natural language processing and text mining. Its main goal is to detect human opinions that are expressed in natural language, by determining its sentiment. Sentiment means "what one feels about something", "an attitude toward something" or simply "an opinion" [40].

Sentiment identification can be divided into several steps [41]:

- **Textual fragments extraction** - This step is responsible for extracting all the textual fragments that have opinions. These fragments can be of two types: objective fragments, that are pieces of text that express facts, and subjective fragments that are pieces of text that express feelings. Both these fragment types may contain opinions, subjective fragments, though, are most likely to be opinionated.
- **Sentiment identification** - Sentiments are treated, in most cases, as a binary scale of positive and negative. The goal of this phase is then to identify if a fragment as a positive or a negative connotation.
- **Determining overall sentiment** - Finally, when the sentiment is identified for each fragment, the overall sentiment for the whole text must be determined.

Chapter 3

Nutritional Supplement Ontology

The previous chapter has shown the different methodologies available regarding recommendation systems and natural language processing. This one, however, will focus on the nutritional ontology that had to be created for this project. It will begin with a brief theoretical introduction of what an ontology is, followed by different types and languages utilized. Next, the decisions made regarding the creation of the nutritional ontology will be explained, as well as how it was populated.

3.1 What is an ontology?

There are many definitions for the word "ontology". However, according to [42] the term can be distinguished between its usage as an uncountable or a countable noun. The first case refers to the usage of Ontology (with uppercase initial), a branch of philosophy that deals with the structure and nature of things. It does not take into account if the thing exists or not, i.e. even though mythical creatures never existed, it becomes plausible to scrutinize their characteristics in order to categorize them and their relations [43]. The latter case is mostly used in Computer Science as a computational artifact. However, this last case does take into account the existence of the "thing", as for what exists is that which can be represented [44].

3.2 Types of ontologies

As stated in [45], there are different types of ontologies, determined by their level of generalization:

- ***Domain ontology*** - holds the knowledge that is valid for a specific type of domain, for example the medical domain.

- ***Metadata ontology*** - supplies a vocabulary for content description of online information sources. A good example of a *metadata ontology* is the *Dublin Core Metadata Initiative* (DCMI).
- ***Generic ontology*** - captures general knowledge about the world. Concepts and notions that are capture are valid across various domains. A well known example of a *generic ontology* is the *CYC* project.
- ***Representational ontology*** - supplies representational entities without disclosing what should be represented. Consequently, this type of ontology does not have any associated domain.
- ***Task and method ontologies*** - both present a reasoning point of view on a domain knowledge. *Task ontologies* are used to supply specific terms for particular tasks while *method ontologies* supply specific terms for problem-solving methods (PSM).

By analyzing all the different ontology types and this project's context, a domain ontology is the most appropriate. The context will be the nutritional supplement recommendation, which is a very specific domain.

3.3 Ontology languages

There are several different languages used to describe an ontology. However, these languages have to meet certain requirements [46], such as:

- Every language should have a compact syntax and be intuitive to humans;
- Should have unambiguous formed semantics and include reasoning properties;
- Should be able to represent human knowledge;
- Have the potential for building knowledge bases;
- Have a proper link with existing web standards to ensure interoperability.

Furthermore, it must be able to describe meaning in a machine-readable way. So, an ontology language has to be able to specify vocabulary and formally define it so that it can work with automated reasoning.

3.3.1 Recent ontology languages

Recently, some ontology languages have been used that have higher utility in the semantic web context. Some of them are the Resource Description Framework and the Ontology Web Language.

Resource Description Framework

The Resource Description Framework, or RDF for short, is a very common markup scheme, being mainly used for machine-processable semantics of data description. The main goal is to add formal semantics to the web and provide a schema for representing the semantics of the data in a standardized manner.

A schema is a model that is used to define how information is structured [47]. In other words, it establishes the possible organization of tags and text in a valid document. The validity of said document indicates that its structure fits within the previously defined schema. Regarding the RDF Schema (RDFS), the same principle applies: it is a model that describes related resources and relationships between them [48]. The tables 3.1 and 3.2 provide a quick overview of the class and property system of the RDFS.

Class Name	Description
rdfs:Resource	The class of everything. All other classes are subclasses of rdfs:Resource.
rdfs:Literal	The class of literal values, i.e. strings and integers.
rdf:langString	The class of language-tagged string values.
rdf:HTML	The class of HTML literal values.
rdfs:Class	The class of resources that are RDF classes.
rdf:Property	The class of RDF properties.
rdfs:Datatype	The class of datatypes, i.e. strings, numbers and dates.
rdf:Statement	The class of RDF statements.
rdf:Bag	The class of unordered containers.
rdf:Seq	The class of ordered containers.
rdf:Alt	The class of containers of alternatives.
rdfs:Container	The class of RDF containers.
rdfs:ContainerMembershipProperty	The class of container membership properties, all of which are sub-properties of 'member'.
rdf:List	The class of RDF lists.

Table 3.1: RDF Schema Classes

Property Name	Description	Domain	Range
rdf:type	The subject is and instance of a class.	rdfs:Resource	rdfs:Class
rdfs:subClassOf	The subject is a subclass of a class.	rdfs:Class	rdfs:Class
rdfs:subPropertyOf	The subject is a sub-property of a property.	rdf:Property	rdf:Property
rdfs:domain	A domain of the subject property.	rdf:Property	rdfs:Class
rdfs:range	A range of the subject property.	rdf:Property	rdfs:Class
rdfs:label	Human-readable name for the subject.	rdfs:Resource	rdfs:Literal
rdfs:comment	Description of the subject resource.	rdfs:Resource	rdfs:Literal
rdfs:member	A member of the subject resource.	rdfs:Resource	rdfs:Resource
rdf:first	The first item in the subject RDF list.	rdf:List	rdfs:Resource
rdf:rest	The rest of the subject RDF list after the first item.	rdf:List	rdfs:List
rdfs:seeAlso	Further information about the subject resource.	rdfs:Resource	rdfs:Resource
rdfs:isDefinedBy	The definition of the subject resource.	rdfs:Resource	rdfs:Resource
rdf:value	Idiomatic property used for structured values.	rdfs:Resource	rdfs:Resource
rdf:subject	The subject of the subject RDF statement.	rdf:Statement	rdfs:Resource
rdf:predicate	The predicate of the subject RDF statement.	rdf:Statement	rdfs:Resource
rdf:object	The object of the subject RDF statement.	rdf:Statement	rdfs:Resource

Table 3.2: RDF Schema Properties

Concerning syntax, RDF can be written utilizing various formats, such as XML, Notation 3 (N3), and Turtle.

The World Wide Web Consortium (W3C) specs were responsible for enabling RDF to be written in XML format, also called RDF/XML [49]. Documents created in that format, are comprised of two distinct node types: resource and property nodes. The resource nodes are the subjects and objects of statements and commonly have an attribute with the URI of the resource they represent. Moreover, these types of nodes can only contain property nodes, which represent different statements. On the other hand, property nodes can contain literal values, a reference to another resource, or even a full resource node. A brief example of RDF/XML is shown in 3.3.

```
<?xml version="1.0"?>
   <rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
            xmlns:si="https://www.w3schools.com/rdf/">
       <rdf:Description rdf:about="https://www.w3schools.com">
         <si:title>W3Schools</si:title>
         <si:author>Jan Egil Refsnes</si:author>
       </rdf:Description>
   </rdf:RDF>
</xml>
```

Table 3.3: RDF/XML example [50]

The previous example contains two different statements:

Subject	Predicate	Object
https://www.w3schools.com	si:title	"W3Schools"
https://www.w3schools.com	si:author	"Jan Egil Refsnes"

Table 3.4: Example 3.3 corresponding statements

As can be seen in Table 3.4, the hierarchical levels of XML are lost when translating into triples, which indicates that hierarchy is not encoded in the RDF.

Notation 3 (N3) and Turtle are also systems used for writing RDF. In contrast to RDF/XML, the main difference is the readability. In these, statements are composed by the subject URI, followed by the predicate URI, supplanted by the object URI or literal value, and finally by a period [49]. Multiple statements with the same subject can additionally be grouped together by using a semicolon and excluding the subject a second time. The same information in Table 3.3 written in N3 and Turtle are written in Table 3.5. In this case, both N3 and Turtle are written in the same way.

OWL

Another very commonly used ontology language is the Web Ontology Language (OWL). OWL was designed to represent intricate knowledge about anything, and groups of and relations between things [51].

```
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix si: <https://www.w3schools.com/rdf/> .

<https://www.w3schools.com> si:title "W3Schools" ;
                            si:author "Jan Egil Refsnes" .
```

Table 3.5: N3 and Turtle example.

Similar to RDF, OWL has multiple syntaxes available:

- **RDF/XML** - RDF/XML syntax with a distinct version for the OWL constructs.
- **Manchester Syntax** - OWL syntax designed for easier reading.
- **OWL XML** - Syntax defined by and XML schema.

OWL can also be, similarly to RDF, serialized in Turtle format, when using the RDF-based syntax.

Information is expressed in OWL by use of three simple notions: axioms, entities, and expressions. Axioms are the basic statements, or propositions, that an ontology represents, and can be either true or false. Entities are elements used to point to real-world things. Finally, expressions are the distinct compounds of elements that form elaborate descriptions.

Contrary to RDF, OWL is not a schema language. Therefore, it does not have specific schema classes and properties. However, it may have classes and properties specific to the context of the ontology. For instance, an ontology about a company might have classes such as departments and employees, and properties - object or datatype properties - such as an employee that works in a certain department and the employee name.

3.4 Source of data

Contrary to expectations, there are not many nutritional supplement datasets available. The United States National Library of Medicine and the Office of Dietary Supplements, however, released all the information available on the American nutritional supplements obtainable on the market [52].This dataset contains 65,684 different products, each with their own respective ingredients and labels.

The data acquisition task is done in two separate steps. The first one is downloading the available product information from the DSLD official website. This data comes in a JSON file that is posteriorly parsed. The final step is to make HTTP requests to the DSLD API in order to obtain the product label statements and their respective ingredients. Both of these steps will be further explained in section 3.6.

3.5 Nutritional ontology structure

The first step of creating the natural supplement ontology was to define its syntax and structure. First of all, using XML as the syntax for RDF is recommended. However, XML syntax has some drawbacks; some predicate URIs are prohibited and XML 1.0 prevents the encoding of some Unicode codepoints. Turtle syntax, on the other hand, does not possess these constraints [53]. The solution was then assayed for the RDF ontology using the Turtle syntax.

Once the ontology syntax was defined, it was then necessary to establish it's structure. By examining the dataset, two main classes of objects are distinguished: Supplement and Ingredient. Both of these classes contain subclasses that further specify the available types of objects, as it is shown in Tables 3.6 and 3.7, respectively.

Class	Subclass	Sub-subclass
Supplement	Dietary	Amino Acid / Protein
		Botanical
		Combination
		Other Combination
		Diet Mineral
		Diet Vitamin
		Herbal
		Non-Nutrient
		Other Nutritive
	Fatty Acid	-
	Multi-Mineral	-
	Multi-Vitamin	-
	Single-Mineral	-
	Single-Vitamin	-

Table 3.6: Supplement subclasses

Class	Subclass
Ingredient	Amino Acid
	Animal Part
	Bacteria
	Blend
	Carbohydrate
	Chemical
	Enzyme
	Fat
	Fiber
	Header
	Hormone
	Mineral
	Other Ingredient
	Polysaccharide
	Protein
	TBD (to be determined)
	Vitamin

Table 3.7: Ingredient subclasses

Supplements and ingredients have specific characteristics that ought to be considered. First of all, in an ontology, there are two different types of properties [54]:

1. **Object properties** - These are properties that link individuals to other individuals, i.e., if a Father-Son relationship is to be represented, then the Son individual has a link to a Father individual.

2. **Datatype properties** - These are properties that link individuals to values, i.e., every Person individual should have a name value, for instance, John.

Generally, all supplements have the same properties. Consequently, supplements will have one object property and nine datatype properties. Such properties are shown in Tables 3.8 and 3.9, respectively.

Object property	**Description**
Ingredient	A supplement has various ingredients, therefore, should be linked to ingredient individuals.

Table 3.8: Supplement class object properties

Data property	Description
Brand name	The brand of the product.
Product name	The name of the product.
Net content	The amount of the product, as declared on the label.
Net content unit	Measurement unit of the product quantity.
Serving size quantity	The amount of product that should be consumed.
Serving size unit	Measurement unit of the serving size.
Statements	Warning and advisory statements that are present on the product label.
Suggested usage	Recommended usage of the product.
Target group	Minimum target age group for the product.

Table 3.9: Supplement class data properties

Ingredients, on the other hand, only have one object and one datatype property, presented in Tables 3.10 and 3.11 accordingly.

Object property	Description
Supplement	An Ingredient individual is part of a Supplement individual, therefore, should be linked to the respective Supplement individual.

Table 3.10: Ingredient class object properties

Data property	Description
Ingredient name	Name of the ingredient.

Table 3.11: Ingredient class data properties

The beforementioned structure and characteristics were all completed using the software named Protégé [1]. Protégé, as described by its authors, is a free, open-source ontology editor that allows users to build knowledge-based solutions in areas, such as nutritional supplementation. In order to further help represent the ontology structure, the ontology graph is shown in Image 3.1.

3.6 Populating the ontology

Including nutritional supplement information in the ontology is of extreme importance. Although ontologies do not work as databases, it is possible to store data the same way a database does. However, there are some notable differences: in a traditional database, if some fact is missing, it is usually considered false. This is called the closed-world

[1] https://protege.stanford.edu/

assumption. On the other hand, if the fact is not present in an ontology, it may only be absent, with the likelihood of it being true, in other words, an open-world assumption.

For the purpose of data acquisition, a JSON file was obtained from the DSLD database. This JSON includes the basic data of all products accessible on the American marketplace. Table 3.14 shows an example of the JSON structure for one product.

After the first portion of information gathering is completed, a Java program was developed. This program parsed the JSON data by using a package [55] dedicated to parsing this kind of format. This parser extracts information of the supplement, such as the product identifier, brand and name, net content and net content unit, serving size quantity and unit, and supplement form. By making use of Object-Oriented programming, it becomes straightforward to group data into class objects.

Subsequently, the program makes an API call for each product parsed previously and retrieves the supplement target group, label statements, and respective ingredients. Likewise, an Ingredient class is created in order to better store data. Finally, it is necessary to create individuals to add to the ontology. In order to so, a library named Apache Jena was used. As the creators claim [56], Apache Jena is a free and open source framework, in Java, used to build Semantic Web and Linked Data applications.

Once the core information is all gathered, it is necessary to build individuals and add their corresponding properties. Firstly, the objects were classified according to the supplement and ingredient subclasses, in order to better separate the different individuals in the ontology. Then, the properties that each object contained were added to the respective individual. Ultimately, relationships were added between the supplement and ingredient individuals so that they better express the real-world scenario.

The last step of this process is, finally, to create and save a Turtle ontology model with all the individuals previously created. As it can be seen in Tables 3.12 and 3.13, there are 65,684 and 47,941 different supplement and ingredient individuals, respectively. Also, the subclass Dietary is not being instantiated. This subclass is comprised of 9 distinct subclasses. Although it has no practical use in the system, this class is present just as a way of organizing the different supplement taxonomies.

This model is later used in the recommendation system. The Image 3.2 is a visual representation of the pipeline being used in this process.

Class Name	Number of individuals
Amino Acid / Protein	2,713
Botanical	9,799
Combination	0
Other Combination	23,507
Diet Mineral	7,031
Diet Vitamin	3,386
Herbal	9,799
Non-Nutrient	6,693
Other Nutritive	353
Fatty Acid	2,403
Multi-Mineral	0
Multi-Vitamin	0
Single-Mineral	0
Single-Vitamin	0

Table 3.12: Number of supplement individuals per class

Class Name	Number of individuals
Amino Acid	2,210
Animal Part	912
Bacteria	2,330
Blend	6235
Carbohydrate	233
Chemical	6,604
Enzyme	2,907
Fat	3,103
Fiber	573
Header	1,381
Hormone	77
Mineral	8,349
Other Ingredient	5,300
Polysaccharide	119
Protein	582
TBD (to be determined)	0
Vitamin	7,026

Table 3.13: Number of ingredient individuals per class

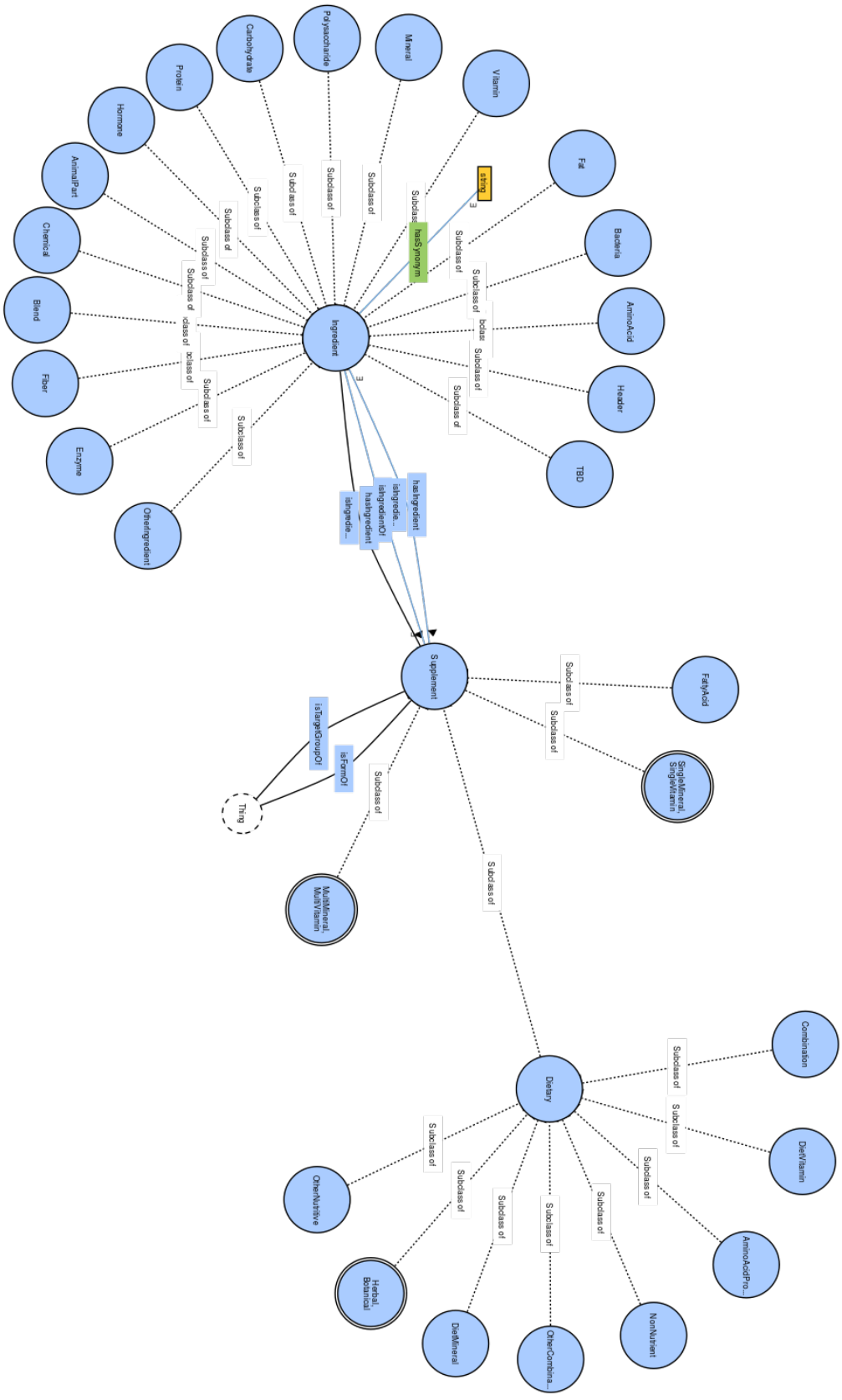

Figure 3.1: Ontology graph structure

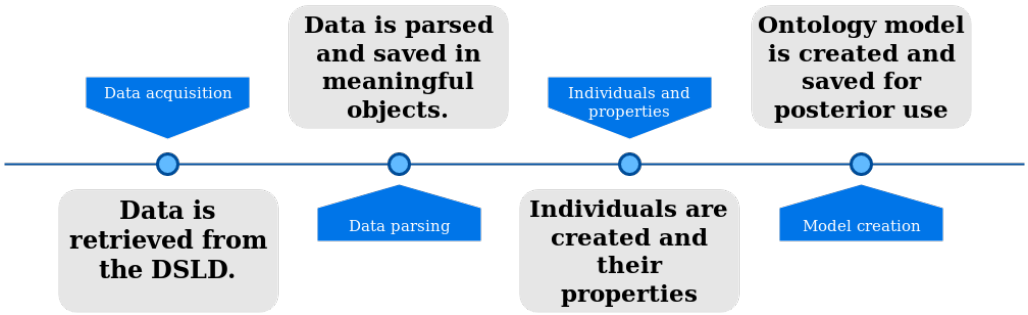

Figure 3.2: Ontology creation pipeline


```
{
   "data":[
      [
         "row-2gvn_apue~fn58",
         "00000000-0000-0000-1673-C2B2699C028A",
         0,
         1551234745,
         null,
         1551234745,
         null,
         "{ }",
         "65060",
         "1 Up Nutrition",
         "1 Up Nutrition 1 Up Whey Chocolate & Peanut Butter Blast",
         "2.06",
         "lbs",
         "1",
         "Scoop(s)",
         "DIETARY SUPPLEMENT, AMINO ACID OR PROTEIN [A1305]",
         "POWDER [E0162]",
         "OTHER INGREDIENT- OR CONSTITUENT-RELATED CLAIM OR USE [P0115];
            STRUCTURE/FUNCTION CLAIM [P0265]",
         "HUMAN CONSUMER, FOUR YEARS AND ABOVE [P0250]",
         "adsld",
         "On Market in DSLD",
         "2016-09-22T00:00:00",
         null
      ]
   ]
}
```


Table 3.14: Example of a product in JSON format.

Chapter 4

Methodology

The previous chapter presented the nutritional supplement ontology structure. Furthermore, it also discussed the pipeline for populating such ontology. In this chapter, however, the implementation details are going to be explained. The system architecture is going to be displayed and explained as well.

4.1 System architecture

The system architecture follows a monolithic architecture pattern. Software that follows this type of architecture pattern is designed to be autonomous, where the various elements of the program are connected, thus requiring all the different components to be present in order for the code to compile and run. A diagram depicting the application architecture is shown in Figure 4.1. This type of architecture pattern tends to facilitate development and deployment, as the current integrated development environments support the development of software following this pattern and compile the program into a single file in order to run it. It also facilitates scaling, since various copies of the program can be run at the same time using a load balancer.

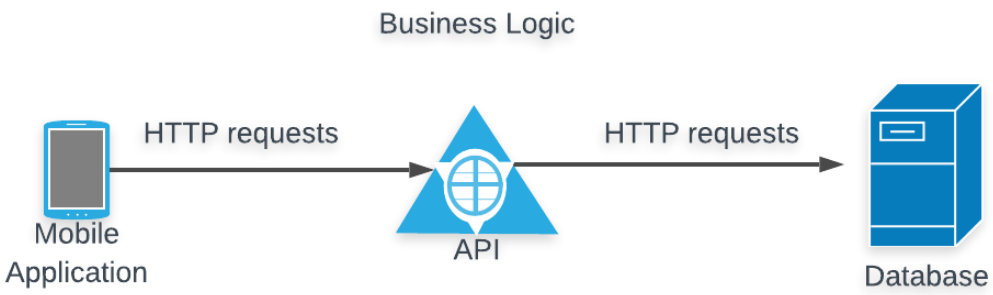

Figure 4.1: Architecture of the system

As it can be seen from the previous figure, the system is divided into three main structures: mobile application, API and database.

4.1.1 Mobile application

The user interacts with the recommendation system via a mobile application. It has the following use cases:

Use case	**Description**
Enter the application	The user clicks the button on the main page to enter the application
Search for supplements	The user searches for supplements in the list
Enter supplement page	The user enters the supplement page and see its details
View recommendations	The user views the recommended supplements, and enter its page

Table 4.1: Application use cases

The application's flow is as follows: the user enters the application, searches for a supplement either by scrolling through a list or by typing its name, selects it, and then the supplement's details will be presented, as well as the most similar supplements. Figure 4.2 shows a visual representation of the application flow.

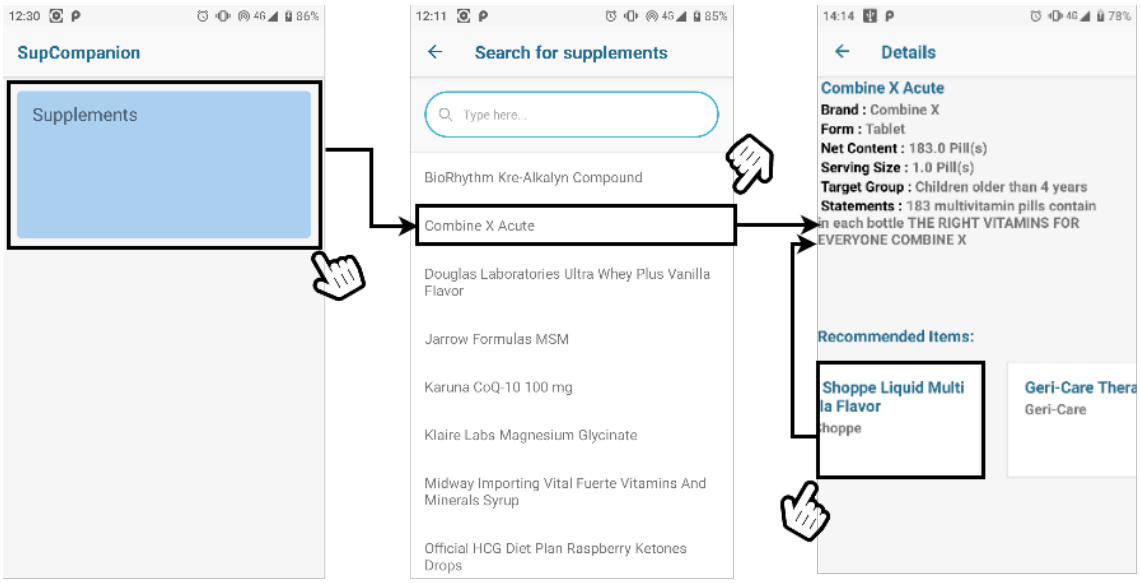

Figure 4.2: Application flow

Supplement search engine

The user can search for supplements. In order to so, the user can either write the full name of the supplement (if it is known) or partial names. By taking advantage of the full text search available on the Neo4j database, the consumer can more easily seek new supplements. By taking Figure 4.2 as an example, if the user either writes "Combine X Acute" or "Cmbine x acte", it will be presented with the same supplement.

The query is sent to the database via an API, that is responsible for processing all the data. In Section 4.1 the API is going to be explained with more detail.

Recommended items

The generated recommendations for the supplements are shown in a horizontal list. The user can scroll through it and, if desired, can press one of the recommendations. This press will redirect the user to the respective supplement, where it can view its information.

The recommendations are always computed when a supplement is created, by sending to the API the name of the supplement. The results are then processed and presented to the user.

Design patterns

The design patterns employed in this application were first used for the LIFANA project [1]. These patterns, although not completely validated, aim to improve the usability of the application, by simplifying the application's learnability, efficiency and reliability [57].

[1] https://www.aicos.fraunhofer.pt/en/our_work/projects/lifana.html

The mobile application, developed using React-Native [2], is responsible for retrieving the information, available in the ontology, using an API. In the application, there is no data processing being done.

4.1.2 Database

Neo4j [3] is a native graph database that provides full database characteristics, including ACID-compliant transactional backend, cluster support, and runtime failover. It also implements the property graph model. Finally, it has some useful features: is utilizes a declarative query language called Cypher, that is very similar to SQL but optimized for graphs; it has constant time traversals in big graphs for both depth and breadth due to efficient representation of nodes and relationships; and it has a flexible property graph schema that can adapt over time.

With that in mind, in this research, a Neo4j graph database is used to store the ontology.

4.1.3 API

An API was created in order to facilitate the dispatch of information between the application and the database. It developed using Flask [4] for the particular reason that it is a micro-framework for Python, which is the language that is used for processing and generation of recommendations. With that in mind, there are five different endpoints for this API. These endpoints are listed and described in Table 4.2.

URL	HTTP verb	Result
/supplements	GET	Returns all supplement entries
/supplements/<text>	GET	Performs a full-text-search with the <text>value and returns all matched supplement entries
/supplement/<node_id>	GET	Returns a supplement entry with the id <node_id>
/supplement/recommendations/<node_id>	GET	Returns a list of generated recommendations for a supplement with the id <node_id>

Table 4.2: API endpoins

4.2 Recommendation System

As it is stated in Section 2.2, a recommendation system collects information on items, processes it, and finally generates predictions. Subsequently, and taking into account this dissertation's domain, the main source of supplement information comes from their statements. As it is explained in Table 3.9, a product statement is the warning and advisory statements that are present on the product label. Since there are no user ratings

[2] https://facebook.github.io/react-native/
[3] https://neo4j.com/
[4] http://flask.pocoo.org/

for the supplements, the best approach for information gathering is the implicit feedback, despite having lower accuracy.

It is not possible to build a collaborative filtering recommendation system without user profiles. However, by using the concepts of the item-based collaborative filtering technique, it becomes possible to create a new engine that is a variation of this method. Both engines first compute a similarity matrix between all pairs of items. The item-based system uses the most similar items that a user has rated to generate a list of recommendations. The new engine, though, creates a list of recommendations by only taking into account the most similar items.

4.2.1 Data pre-processing

The recommendations are generated based on the information that is extracted from the supplement's statements. The statements do not have any structure whatsoever. In addition, there is a possibility that these textual descriptions might contain numbers, special characters and identical words with different case variants. In order to remove this noise, two different steps were employed:

1. **Lower case every word** - The system might handle differently a word in the origin of a sentence with a capital letter from the same word, which shows later in the sentence, but in lower case. This process might lead to a decline in the recommendation accuracy. However, by lowering the case of every single word, the system minimizes this error and increases the overall recommendation accuracy.

2. **Removal of numbers and special characters** - Although numbers are used to describe ingredient quantities within the supplement, they do not add any relevant information for recommendation generation. The same occurs with special characters. By removing said characters and numbers from each separate sentence, there is a considerable noise reduction as well as an increase in accuracy.

After all the supplement statements are pre-processed, the system will proceed to generate summaries of said descriptions.

4.2.2 Statement summarization

As in any text, there are sentences within the statements with higher importance than others. Usually, these sentences are the ones that better characterize the supplement. Therefore, it is paramount to highlight these phrases, in order to create a summary of the statements.

As it is introduced in Section 2.3.2, there exists a document summarization model. This model is called TextRank. Each statement is then split into various sentences. With this set of phrases, it is necessary to compare each pair in order to compute their similarity.

This similarity is calculated by computing the cosine distance between the set of words of each phrase. A similarity matrix is created, and with it, a sentence graph can be created to feed the TextRank algorithm. Subsequently, each node of the graph is then assigned a similarity score.

After the creation of this similarity graph, it is sorted and the top N footnoteIn this case N=5, although the algorithm is prepared to choose a different number. nodes are chosen. With these top N phrases, the statement summaries are now created.

Sentiment Analysis

In order to further enhance the summaries, one could apply sentiment analysis algorithms to the texts, in order to further highlight important phrases. However, and taking into account the descriptive nature of the supplement statements, it was decided that this type of analysis would not benefit the results. This was due to most statements presenting the information in a very neutral manner. Furthermore, even with textual pre-processing, there was still some textual noise. So, when sentiment analysis algorithms are applied, there would be no favourable outcomes.

4.2.3 Generation of recommendations

The final step for this approach is to generate the appropriate recommendations for the supplements. In order to do so, a TF-IDF matrix is created for each summary previously generated. Along with the usage of English stop words, irrelevant tokens that made it through the pre-processing stage are removed. A cosine similarity matrix is once again computed by comparing every pair of supplements that are used in the dataset. With this matrix, by choosing the most similar items, a recommendation list is created. With the purpose of not overflowing the user with an excessive amount of recommendations, some of which might not even be relevant, a limit was set to only show up to ten recommendations.

4.2.4 System pseudo-code

The corresponding pseudo-code for this approach is shown in Table 4.3

```
# DATASET RETRIEVAL #
SET dataset to CALL retrieve_dataset()

# CLEAN UP #
FOR each statement in dataset
    CALL toLowerCase with statements
    CALL removeNumbersAndSpecialCharacters with statement
END FOR

# TEXT RANK #
SET sentence_similarity_matrix to CALL build_similarity_matrix with
    statements, english_stop_words
SET sentence_similarity_graph to CALL graph_from_matrix with
    sentence_similarity_matrix
SET scores to CALL pagerank with sentence_similarity_graph
SET ranked_sentences to CALL sort_top_N_sentences with scores

# TF-IDF ALGORITHM #
SET tf_idf to CALL TfidfVectorizer with english_stop_words
SET matrix to CALL tfidf with summaries
SET similarity_matrix to CALL cosine_similarity with matrix, matrix

# RECOMMENDATION GENERATION #
SET recommendations to CALL get_recommendations with similarity_matrix,
    supplement
```

Table 4.3: Pseudo-code of the implemented code for the item-based collaborative filtering variation in this project. (Source [1]).

Chapter 5

Validation and results

This system was validated regarding the recommendations. In this chapter, such validations are going to be exhibited and justified. Finally, the project results are going to be presented.

5.1 Evaluation metrics

By looking at the recommendations as a classification problem, by using classifier evaluation metrics such as precision and recall [58], the system's performance can be measured.

Precision is defined as being the quotient of the number of true positives by the sum of the number of true positives with false positives. It measures the quantity of correct classifications done by a system. Recall, on the other hand, estimates the number of correct classifications done by the system. It is calculated by dividing the amount of true positives by the sum of true positives with false negatives.

For the purpose of calculating the previous metrics, the outcomes from the classification task must be defined. Within this context, two different classes are being considered, these being Relevant and Irrelevant. With that said, the outcomes are defined in Table 5.1:

The main goal is to maximize the number of relevant recommendations done to the user. However, the number of correct recommendations should also be maximized. In classification tasks it is not possible to optimize both recall and precision, although it

Outcome	**Description**
True Positive (TP)	We recommend a relevant supplement
True Negative (TN)	We do not recommend a supplement that is actually not relevant
False Positive (FP)	We recommend a supplement, but it is actually not relevant
False Negative (FN)	We do not recommend a supplement that is actually relevant

Table 5.1: Different outcome definitions

is possible to compute a new metric that is a blend of these two. This metric is called f1-score and is the harmonic mean of precision and recall. It is calculated by resorting to the following formula:

$$f1-score = 2 * \frac{precision * recall}{precision + recall}$$

5.2 Validation

The classifications were manually done for a set of 50 different supplements and their respective generated recommendations. All the statements were compared by hand and assigned their class, regardless of their similarity score. After the classification task is finished, the different classifications were tallied depending on their scores. Since these were very granular, it was decided that it was best to round the scores to two decimal places in order to count the classes. The result is shown in Figure 5.1.

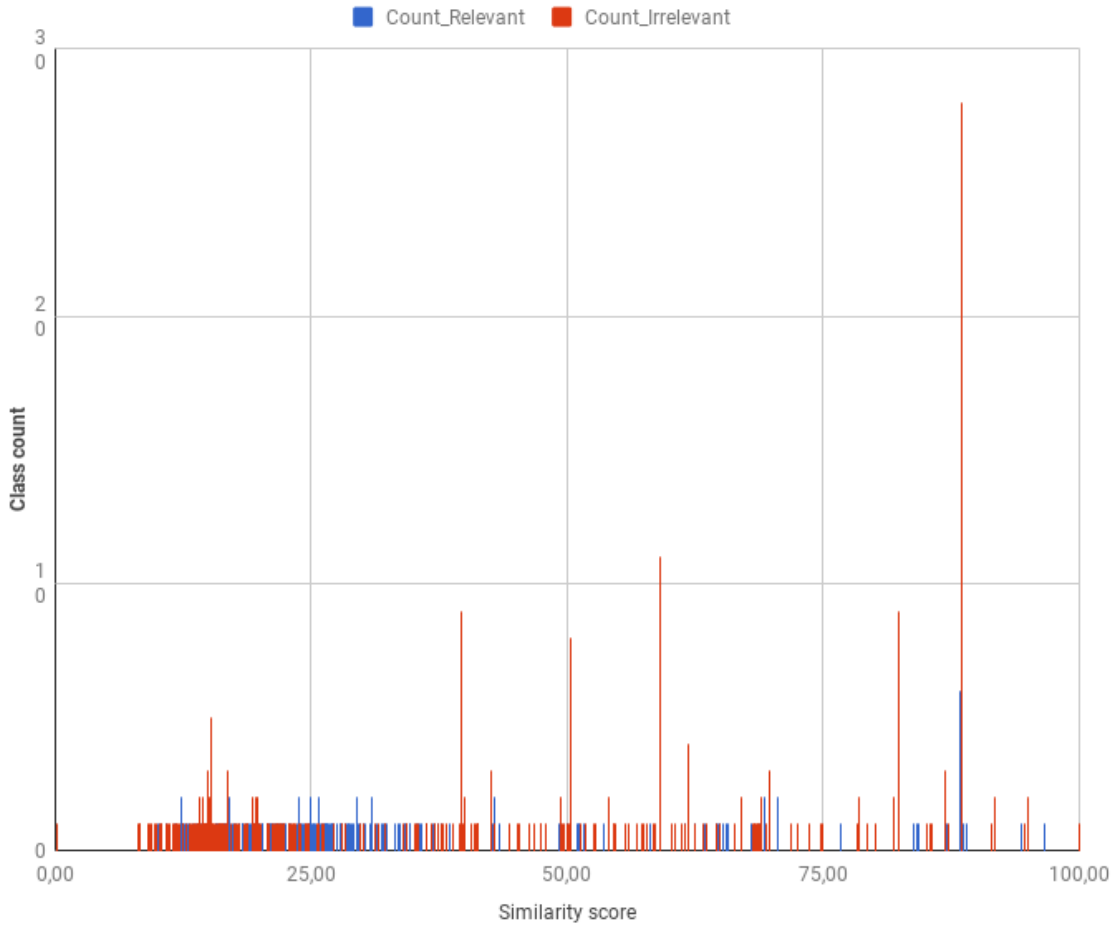

Figure 5.1: Number of individuals of both Relevant and Irrelevant classes.

Following this step was the counting of the number of TP, FP and FN, in order to subsequently calculate all the different evaluation metrics. However, to do so, it is necessary to define a similarity threshold. This limit will define which recommendations can be considered a TP, FP or FN.

A proper recommendation is, ideally, identified by a high similarity score. On the other hand, if the threshold is set too high, the number of recommendations will decrease as the number of available supplements to match do as well. Therefore, the limit should not be too large. For the sake of testing, the threshold was set to 70%.

The peak found on the previous figure is quite interesting: with similarity score close to 90%, there is a count of around 28 Irrelevant classifications. This is a sign that the number of False Positives is very high and that the algorithm results were faulty. Another interesting aspect of the chart is the high density of Relevant classifications in low similarity scores.

An example of a TP, FP and FN is shown in Tables 5.2, 5.3 and 5.4, respectively, and the Table 5.5 shows the different evaluation metric values depending on the threshold used.

	Supplement	Recommendation
Name	Cellucor Alpha Amino Lemon Lime	Cellucor Alpha Amino Grape
Statements	performance aminos sports drink powder with aminos naturally flavored servings . settling may occur	performance aminos sports drink powder with aminos naturally artificially flavored servings settling may occur.
Similarity Score (%)	96,726	
Classification	Relevant	

Table 5.2: Example of a classification true positive

	Supplement	Recommendation
Name	Rexall Natural Fish Oil 1200 mg	Sundown Naturalist Omega 3-6-9
Statements	actual size n a quality guaranteed lab tested laboratory tested to meet strict quality control standards for potency purity and disintegration. since prod. quality guaranteed. b b faa heart health . no	actual size quality guaranteed lab tested laboratory tested to meet strict quality control standards for potency purity and disintegration. quality guaranteed. b b . no. prod
Similarity Score (%)	91,636	
Classification	Irrelevant	

Table 5.3: Example of a classification false positive

	Supplement	Recommendation
Name	NOW Melatonin 1 mg	Indiana Botanic Gardens Valerian Root 500 mg
Statements	research indicates that melatonin may be associated with the regulation of sleep wake cycles. please recycle. melatonin is a potent antioxidant that defends against free radicals and helps to support glutathione activity in the neural tissue actual size helps regulate sleep cycle two stage releasewith co factor nutrients code v made in the u s a.	sleep support
Similarity Score (%)	31,0686	
Classification	Relevant	

Table 5.4: Example of a classification false negative.

Threshold (%)	Precision	Recall	F1
0	0,38	1	0,55
5	0,39	1	0,56
10	0,39	1	0,57
15	0,41	0,89	0,56
20	0,41	0,76	0,53
25	0,42	0,58	0,49
30	0,35	0,38	0,36
35	0,3	0,29	0,29
40	0,28	0,23	0,25
45	0,27	0,21	0,24
50	0,28	0,19	0,22
55	0,27	0,16	0,2
60	0,28	0,15	0,19
65	0,27	0,12	0,17
70	0,26	0,09	0,13
75	0,22	0,07	0,1
80	0,23	0,07	0,1
85	0,24	0,06	0,1
90	0,19	0,04	0,06
95	0,2	0,03	0,06
100	0,18	0,03	0,05

Table 5.5: Different evaluation metric values depending on the used threshold

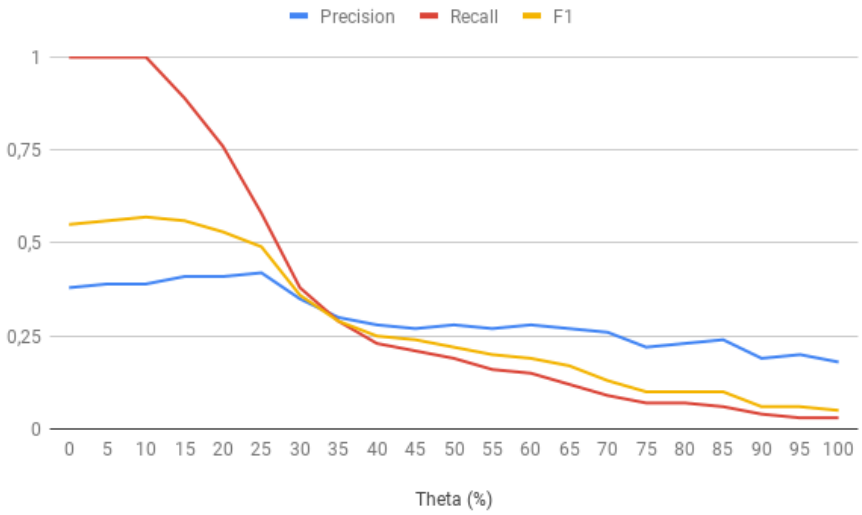

Figure 5.2: Evolution of precision, recall and f1-score depending on the threshold used.

As it can be seen in Table 5.5 and Figure 5.2, if the threshold has the value of 70%, as initially discussed, the f1-score value is very small. Further, in Section 5.3, the main reasons for this low value are discussed. Ultimately, taking into consideration the objective of maximizing the value of f1-score, the used threshold will be 10%. A comparison between the confusion matrices for both thresholds are shown in Images 5.3 and 5.4.

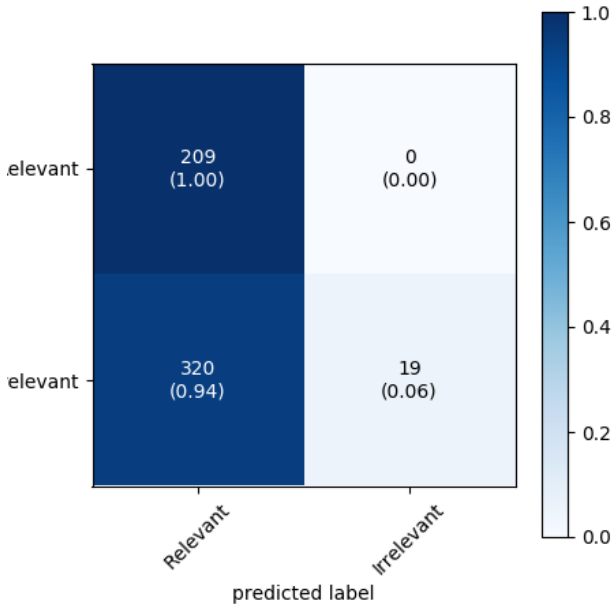

Figure 5.3: Confusion matrix for threshold 10%

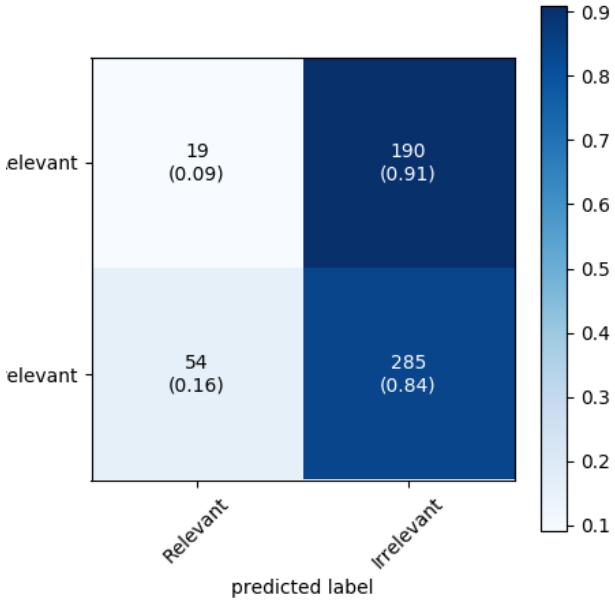

Figure 5.4: Confusion matrix for threshold 70%

5.3 Results

The results obtained from the recommendation system were as expected. Although the used dataset was composed of various supplements, a considerable amount of them did not have proper statements. The DSLD includes all information available on the product label, however, when it is missing it is because the U.S Government does not require the complete data on the supplement. As a consequence of this lack of information, the extracted keywords will not be meaningful for the recommendations and, therefore, the predictions will not be appropriate.

Manual validating a sample set of supplement recommendations made it possible to observe that some similarity scores were very low, yet the recommendations were in fact relevant. There are two main reasons for this incoherence. Firstly, the statements may contain some ambiguous expressions that the TextRank algorithm cannot identify as being important. If this method ranks important sentences as being irrelevant, they will not be added to the summary and, therefore, remove relevance to the recommendation. Finally, one of the TF-IDF algorithm limitations is that it does not take into account the semantics of words and their position in the sentences. For instance, for this algorithm, words like cardiovascular and circulatory have different meanings, even though for us humans they have the same connotation.

As for the mobile application, the results were satisfactory. The main use cases were correctly implemented and the usability design patterns were followed. The final application was developed with the possibility of adding more use cases in mind.

Chapter 6

Conclusions

The goal of this dissertation was to develop a a system capable of informing and recommend nutritional supplements to its users. Prior to the creation of the system, a comprehensive study of the available methodology regarding recommendation systems and natural language processing was done.

The developed engine did not meet the expectations that were originally determined for this dissertation. It was originally expected that the system would be able to successfully provide relevant nutritional supplement recommendations to the user. Although some recommendations are in fact relevant, there is a great amount of them that are not relevant. Such is explained by the need to lower the true positive identification threshold, in order to maximize the f-measure. In this particular case, the threshold is 10% and the f-measure 0,57. It was also possible to conclude that more structured supplement information is needed to correctly generate the recommendations.

Despite that, the work developed in this dissertation can still be adjusted to a variety of different recommendation filtering techniques, in addition to serving as a basis for new applications in the nutritional supplement domain.

6.1 Future work

As previously stated, this system has some limitations. In order to address these shortcomings, some work can be pursued in order to minimize them. Following, is a list of some those efforts:

Ontology improvement

Although the created ontology was successfully implemented, it can still be improved. One enhancement would be to create a recommendation relationship between nodes, in order to minimize processing time and avoid repetition of the same process. Furthermore,

as the number of supplement data sources increases, so should the available information on the ontology. One of the advantages of using an ontology is that different ontologies can merge into one, so, by taking advantage of such benefit, one can create an even more comprehensive nutritional supplement knowledge base.

Dataset improvement

The DSLD dataset is quite comprehensive, containing a large number of nutritional supplements available on the market. However, the information about each of the supplements does not follow any specific structure and is not always present. An improvement on this dataset would be to uniform the texts in order to facilitate the keyword extraction phase, or even create the system's own dataset, by manually adding supplements and transcribing their statements. Although it is an arduous task, it guarantees that each supplement has structured information, and consequently improve the quality of recommendations.

Validation

The validation of the sample set of recommendations was done by someone with no nutritional background. An enhancement for such validations would be to request the assistance of a professional nutritionist, in order to correctly validate the recommendations.

Word embedding

In order to improve the accuracy of the feature extraction from the supplement statements, one can use word embeddings. This technique is capable of capturing word context in a document, as well as the semantic and syntactic similarity. This will greatly improve the accuracy of the system, since it removes the limitations of the TF-IDF algorithm. There are already various word embedding models that can be used, such as GloVe [1], word2vec [2] and fastText [3]. However, it is still possible to create a model specifically for the nutritional domain.

Incremental learning

Another possible improvement for this system is to create a platform where the recommendations can be evaluated as being relevant or irrelevant. By doing this, and by saving the user input, it becomes possible to create a training model without having to manually annotate each of the nutritional supplement statements.

[1] https://nlp.stanford.edu/projects/glove/
[2] https://code.google.com/archive/p/word2vec/
[3] https://fasttext.cc/

Application

The recommendation system was successfully integrated into the mobile application. Nonetheless, more use cases can be added. The database contains information about the ingredients present on the various nutritional supplements. This data can be used to include a search engine that retrieves the supplements that contain a certain ingredient.

Bibliography

[1] George-Bogdan Ivanov. TextRank for Text Summarization. https://nlpforhackers.io/textrank-text-summarization/. [Online; accessed May, 2019].

[2] Colin W. Binns, Mi Kyung Lee, and Andy H. Lee. Problems and Prospects: Public Health Regulation of Dietary Supplements. *Ssrn*, 2018.

[3] Bimal H. Ashar and Anastasia Rowland-Seymour. Advising Patients Who Use Dietary Supplements. *American Journal of Medicine*, 121(2):91–97, 2008.

[4] Don Gilbert. Intelligent Agents : The Right Information at the Right Time 2 . Problems Intelligent Agents Can Solve 3 . Scenarios of Intelligent Agent Use. pages 1–9, 1997.

[5] Barbara Hayes-Roth. An architecture for adaptive intelligent systems. *Artificial Intelligence*, 72(1-2):329–365, 1995.

[6] Utkarshraj Atmaram. Example of an intelligent agent. https://commons.wikimedia.org/wiki/File:IntelligentAgent-SimpleReflex.png, 2006. [Online; accessed January, 2019].

[7] John Nealon and Antonio Moreno. Agent-Based Applications in Health Care. *in Applications of Software Agent Technology in the Health Care Domain , Whitestein Series in Software Agent Technologies, Birkhäuser Verlag*, i:3–18, 2003.

[8] Sajid Iqbal, Wasif Altaf, Muhammad Aslam, Waqar Mahmood, and Muhammad Usman Ghani Khan. Application of intelligent agents in health-care: review. *Artificial Intelligence Review*, 46(1):83–112, 2016.

[9] G Eysenbach. What is e-health? *Journal of medical Internet research*, 3(2):20, 2001.

[10] Martin J. Eppler and Jeanne Mengis. The concept of information overload: A review of literature from organization science, accounting, marketing, MIS, and related disciplines. *Information Society*, 20(5):325–344, 2004.

[11] C. Porcel, J. M. Morales Del Castillo, M. J. Cobo, A. A. Ruiz, and E. Herrera-Viedma. An improved recommender system to avoid the persistent information overload in a university digital library. *Control and Cybernetics*, 39(4):898–923, 2010.

[12] Robin Burke. Hybrid Recommender Systems: Survey and Experiments. pages 1–29.

[13] Duen Ren Liu, Chin Hui Lai, and Wang Jung Lee. A hybrid of sequential rules and collaborative filtering for product recommendation. *Information Sciences*, 179(20):3505–3519, 2009.

[14] F. O. Isinkaye, Y. O. Folajimi, and B. A. Ojokoh. Recommendation systems: Principles, methods and evaluation. *Egyptian Informatics Journal*, 16(3):261–273, 2015.

[15] Kartik Chandra Jena, Sushruta Mishra, Soumya Sahoo, and Brojo Kishore Mishra. Principles, Techniques and Evaluation of Recommendation Systems. pages 1–6, 2017.

[16] Michael J. Pazzani and Daniel Billsus. *Content-Based Recommendation Systems*, pages 325–341. Springer Berlin Heidelberg, Berlin, Heidelberg, 2007.

[17] Prem Melville and Vikas Sindhwani. Recommender Systems. pages 1–21, 2017.

[18] Roberto Saia, Ludovico Boratto, Salvatore Carta, and Gianni Fenu. Semantics-aware content-based recommender systems: Design and architecture guidelines. *Neurocomputing*, 10 2016.

[19] Be-Deu-Ro Kim and Jee-Hyong Lee. Improved post-filtering method using context compensation. *The International Journal of Fuzzy Logic and Intelligent Systems*, 16:119–124, 06 2016.

[20] Zhenxue Zhang. Improving Memory-Based Collaborative Filtering using a Factor-Based Approach. 2007.

[21] Greg Linden, Brent Smith, and Jeremy York. Amazon.com recommendations: Item-to-item collaborative filtering. *IEEE Internet Computing*, 7(1):76–80, January 2003.

[22] Yehuda Koren, Robert Bell, and Chris Volinsky. Matrix factorization techniques for recommender systems. pages 42–49, 2009.

[23] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. Item-Based Collaborative Filtering Recommendation Algorithms.

[24] Emre Sezgin and Sevgi Özkan. A systematic literature review on Health Recommender Systems. *2013 E-Health and Bioengineering Conference, EHB 2013*, (November 2013), 2013.

[25] Andreas Holzinger. Machine learning for health informatics. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 9605 LNCS(November 2017):1–24, 2016.

[26] FitGenie. Fitgenie. https://www.fitgenieapp.com, 2019. [Online; accessed January, 2019].

[27] Fooducate. Fooducate. https://www.fooducate.com, 2019. [Online; accessed January, 2019].

[28] David D. Lewis and Karen Spärck Jones. Natural language processing for information retrieval. *Commun. ACM*, 39(1):92–101, January 1996.

[29] Jim Cowie and Wendy Lehnert. Information extraction. *Commun. ACM*, 39(1):80–91, January 1996.

[30] Sven Hartrumpf, Hermann Helbig, and Rainer Osswald. Semantic interpretation of prepositions for NLP applications. *Computational Linguistics*, (April):29–36, 2006.

[31] Gisela Håkansson, Manfred Pienemann, and Susan Sayehli. Language Processing. *Second Language Research*, 349(6245), 2002.

[32] Why Natural Language Processing (NLP) is a core AI Technology – Witan World. https://witanworld.com/blog/2018/10/28/naturallanguageprocessing-nlp/. [Online; accessed May, 2019].

[33] Samir Prado Daud and Carlos Henrique Costa Ribeiro. NLP - Lexical Analysis applied to requirements. *Proceedings of the 9th Brazilian Conference on Dynamics Control and their Applications*, pages 1091–1098, 2010.

[34] Mallamma V Redd and M Hanumanthappa. Semantical and Syntactical Analysis of NLP. 5(3):3236–3238, 2014.

[35] Cliff Goddard. Semantic Analysis. (January 2010), 2016.

[36] Juan Ramos. Using TF-IDF to Determine Word Relevance in Document Queries Juan. *Zeitschrift für anorganische und allgemeine Chemie*, 164:45–56, 1931.

[37] Charu C. Aggarwal and ChengXiang Zhai. *Mining text data*. 2012.

[38] Rada Mihalcea and Paul Tarau. TextRank: Bringing order into text. In *Proceedings of EMNLP 2004*, pages 404–411, Barcelona, Spain, July 2004. Association for Computational Linguistics.

[39] Sergey Brin and Lawrence Page. The anatomy of a large-scale hypertextual web search engine. *Comput. Netw. ISDN Syst.*, 30(1-7):107–117, April 1998.

[40] Bing Liu. Sentiment Analysis and Opinion Mining. (May), 2012.

[41] Mohsen Farhadloo and Erik Rolland. Fundamentals of Sentiment Analysis and Its Applications. (March), 2016.

[42] Survey Provides, A Conceptual Introduction, and Their Role. Ontology as vocabulary. *Intelligent Systems and their Applications, IEEE*, 14(1):20–26, 1999.

[43] Nicola Guarino, Daniel Oberle, and Steffen Staab. *What Is an Ontology?*, pages 1–17. 05 2009.

[44] Gruber T. Toward principles for the design of ontologies used for knowledge sharing. *International Journal of Human-Computer Studies*, (43):907–928, 1995.

[45] Dieter Fensel. Ontologies: A Silver Bullet for Knowledge Management and Electronic Commerce. *Kybernetes*, 33:1544–1546, 2004.

[46] J. R.G. Pulido, M. A.G. Ruiz, R. Herrera, E. Cabello, S. Legrand, and D. Elliman. Ontology languages for the semantic web: A never completely updated review. *Knowledge-Based Systems*, 19(7):489–497, 2006.

[47] Norman Walsh. What is a schema. https://www.xml.com/pub/a/1999/07/schemas/whatis.html, 1999. [Online; accessed April, 2019].

[48] W3C. Rdf schema 1.1. https://www.w3.org/TR/rdf-schema, 2014. [Online; accessed April, 2019].

[49] Joshua Tauberer. What is rdf. https://www.xml.com/pub/a/2001/01/24/rdf.html, 2006. [Online; accessed April, 2019].

[50] XML RDF. https://www.w3schools.com/xml/xml_rdf.asp. [Online; accessed April, 2019].

[51] Pascal Hitzler and Sebastian Rudolph. OWL 2 Web Ontology Language Primer (Second Edition). Technical report, 2012.

[52] National Library of Medicine Office of Dietary Supplements. Dietary Supplement Label Database (DSLD). http://www.dsld.nlm.nih.gov/dsld/index.jsp. [Online; accessed January, 2019].

[53] Tim Berners-Lee David Beckett. Turtle - terse rdf triple language. https://www.w3.org/TeamSubmission/turtle/, 2011. [Online; accessed April, 2019].

[54] Matthew Horridge, Simon Jupp, Georgina Moulton, Alan Rector, Robert Stevens, and Chris Wroe. A Practical Guide To Building OWL Ontologies Using Protege 4 and CO-ODE Tools. *Edition 1.1*, page p. 102, 2007.

[55] Json in java [package org.json]. https://github.com/stleary/JSON-java, 2015. [Online; accessed in April, 2019].

[56] Apache jena. https://jena.apache.org/index.html, 2011. [Online; accessed in April, 2019].

[57] Natalia Juristo Juzgado, Marta López, Ana María Moreno, and Maria Isabel Sánchez Segura. Improving software usability through architectural patterns. *Proceedings of ICSE 2003 Workshop on Bridging the Gaps Between Software Engineering and Human-Computer Interaction, May 3-4, 2003, Portland, Oregon, USA*, (January):12–19, 2003.

[58] Francesco Ricci, Lior Rokach, Bracha Shapira, Paul B Kantor, and Francesco Ricci. *Recommender Systems Handbook*. 2015.